2002-09

Virtual Close Quarter Battle (CQB) graphical decision trainer

Reece, Jordan

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

VIRTUAL CLOSE QUARTER BATTLE (CQB) GRAPHICAL DECISION TRAINER

by

Jordan Reece

September 2002

Thesis Advisor: Rudy Darken
Co-Advisor: Joseph Sullivan

This thesis done in cooperation with the MOVES Institute.
Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2002	Master's Thesis

4. TITLE AND SUBTITLE: Virtual Close Quarter Battle (CQB) Graphical Decision Trainer

5. FUNDING NUMBERS

6. AUTHOR (S): Jordan D. Reece

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the U.S. Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited	Statement A

13. ABSTRACT *(maximum 200 words)*

The physical skills exhibited by Marines conducting CQB operations are a reflection of their cognitive adaptation and performance while in that environment. While these physical skills require continuous reinforcement for the Marine to remain effective, so does the Marine's ability to make rapid and sound decisions according to Marine Corps CQB Doctrine. During extended periods at sea, Marines are currently afforded little or no practical application of these skills due to a lack of proper facilities aboard ship.

The Virtual CQB Graphical Decision Trainer (VCGDT) fills this void. VCGDT provides an interactive virtual environment where Marines can communicate both verbally and non-verbally; and coordinate training activities and simulated missions. In addition, the system promotes the introduction of team leaders and instructors into the environment for real-time virtual supervision and instruction.

The VCGDT hardware design takes into consideration the size and bandwidth constraints posed by placement on a Naval vessel. Additionally, the VCGDT attempts to circumvent problems associated with some types of tracking systems due to ship composition and radar; and locomotion devices due to excessive ship motion.

14. SUBJECT TERMS: Virtual Trainer, Virtual Environment, Close Quarter Battle, Virtual CQB, Marine Trainer, Dismounted Infantry, Cognitive Training.	15. NUMBER OF PAGES 77
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 **Standard Form 298 (Rev. 2-89)** **Prescribed by ANSI Std. 239-18**

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

VIRTUAL CLOSE QUARTER BATTLE (CQB) GRAPHICAL DECISION TRAINER

Jordan Reece
Captain, United States Marine Corps
B.S., Portland State University, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2002

Author:

Jordan Reece

Approved by:

Rudy Darken, Thesis Advisor

CDR Joe Sullivan, Co-Advisor

Chris Eagle, Chairman
Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The physical skills exhibited by Marines conducting CQB operations are a reflection of their cognitive adaptation and performance while in that environment. While these physical skills require continuous reinforcement for the Marine to remain effective, so does the Marine's ability to make rapid and sound decisions according to Marine Corps CQB Doctrine. During extended periods at sea, Marines are currently afforded little or no practical application of these skills due to a lack of proper facilities aboard ship.

The Virtual CQB Graphical Decision Trainer (VCGDT) fills this void. VCGDT provides an interactive virtual environment where Marines can communicate both verbally and non-verbally; and coordinate training activities and simulated missions. In addition, the system promotes the introduction of team leaders and instructors into the environment for real-time virtual supervision and instruction.

The VCGDT hardware design takes into consideration the size and bandwidth constraints posed by placement on a Naval vessel. Additionally, the VCGDT attempts to circumvent problems associated with some types of tracking systems due to ship composition and radar; and locomotion devices due to excessive ship motion.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENTS

Many people deserve a word of thanks for making this research possible. First, I appreciate the confidence and respect of Dr. Rudy Darken. His support has always been in a form consistent with a mentor and not just an expert. It has been my privilege to work with him. I'd also like to thank the team involved in the production of the gflib project to include Erik Johnson for his expert programming experience, Matt Prichard for his professional models, and CDR Eric Krebs for all of his audio expertise and use of his gfAudio library. Without their help, this project would never have come so far. Most importantly, however, I'd like to thank my wife Tracey and my two boys Brandon and Garrett for their support and most of all patience. It's because of you that I am constantly reminded to seek self improvement. Thank you.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PROBLEM STATEMENT

The Marine Corps has a six to nine month training pipeline prior to a deployment for Marines participating in CQB operations. The training consists of a five week CQB course along with several weeks of collateral CQB training including courses in manual breaching and assault climbing. Following their preliminary training, they participate in a Work-Up cycle designed to evaluate their skills in numerous exercises conducted by the Marine Expeditionary Unit (MEU) to which they are assigned. The exercises test the Marines in various CQB environments and missions including Visit, Board, Search and Seizures (VBSS), In-extremis Hostage Recovery (IHR) and Gas and Oil Platform (GoPlat) scenarios. Following a successful evaluation by members of the Special Operations Training Group (SOTG), the MEU is awarded the title of Special Operations Capable (SOC) and departs on its deployment.

During the deployment phase, these Marines sit aboard ship for extended periods of time with little or no practical application of their skills. Among these skills is the ability to make rapid and sound decisions according to Marine Corps CQB Doctrine. These decisions are vital to mission success in that small decisions by a single Marine may have a large impact on the entire force. Many of these decisions become routine with adequate exposure to a variety of operational environments but become labored by lack of this exposure. Skill atrophy is especially dangerous because it can occur without the conscience

knowledge of the individual Marine. However, it becomes apparent during exercises abroad, after the atrophy has been allowed to erode their ability to effectively accomplish their mission.

B. THESIS OBJECTIVES

A good deal of research has shown that well designed virtual flight simulators have the ability to enhance the training of individual pilots. This thesis study will attempt to achieve the same type of success in a ground-based scenario. A computer-based simulator may be able to provide these Marines an opportunity to immerse themselves in a virtual environment designed to maintain decision-making and doctrinal skills in absence of a real world training facility.

This thesis will suggest that many of the skills subject to degradation are not necessarily effectively modeled, however, many are. It would be the wrong approach to attempt to completely replace the real with the virtual, because of these difficult to model skill-sets. Instead, this thesis hopes to focus on the idea of simulating doctrinal cognitive functions within a CQB environment. It will attempt to show that it's possible to reinforce the training of the less tangible, but equally important skill of rapid decision-making.

C. SCOPE OF THIS THESIS

This thesis has little in the way of prior research with which to build upon. It's because of this that implementation details and ideas must be tested by the production of a prototype. To do this, this thesis will focus on three primary stages of evolution: Requirements

definition, prototype software development, and hardware implementation.

This stage involves deciding on appropriate cognitive and physical implementations for virtual immersion and establishing requirements for the graphical interface. Once the appropriate interface is established, it is integrated into the basic requirements for effective CQB training goals.

The engineered application is built to suit both the needs of the requirements and training goals established in the first stage. Additionally, network and interface issues are refined based on existing constraints for the application's use. Based on software constraints, the hardware is integrated into the system to produce a prototype VCGDT.

D. THESIS ORGANIZATION

The remainder of the thesis is organized into the following chapters:

- Chapter II - Background. This chapter briefly discuses the history of military simulations, describes what virtual training really means, and talks about what types of personal computer (PC) based systems are currently in use. Additionally, it presents information that examines the effectiveness of this training and how effective training in the real world relates to this virtual one.
- Chapter III - Approach. This chapter describes the manner in which the prototype system evolved. It

looks at the types of constraints and considerations that helped shape the design of the system.

- Chapter IV - Implementation. This chapter outlines the C++ classes built to support the prototype application as well as the hardware system considerations which lead to the VCGDT.
- Chapter V - Conclusion. Concludes the study with a general summary and identification of areas for future research.

II. BACKGROUND

A. VIRTUAL TRAINING

1. History of Military Simulation Training

Military simulations have evolved to their current state through a long history of experimentation and subsequent improvement. Models of warfare go back as far as 5000 years ago. In fact historical records show that around 3000BC, the Chinese developed a game they called Wei-Hai, which is thought to be similar to the modern game of Go. Go surfaced around 2200BC.

The game of Chaturanga showed up around 500AD in India as a predecessor to the modern day game of Chess. Although smaller in scale, it allowed for the introduction of two or four players. Chess evolved in Southern Europe around 1400AD and has been used as a soundboard for strategic thinking ever since.

The Roman legions were thought to have used sand tables and miniature battle pieces around 30AD; a concept that has survived the test of time as a tool for training soldiers in military academies and schools. It's only with the advent of computers that we see a gradual trend toward using virtual simulations to replace these traditional tools [**1**].

Virtual training systems have had a long history of association with military training. The most obvious and oldest among them are flight simulators, which are still used today in the routine training of military pilots. Traditionally, these systems were produced as a function of operator tasks specific to an individual's training needs.

Their design facilitated the improvement of individual skill to augment real world training and overcome shortfalls in training resources. Only more recently has the focus turned to the extensible purpose of interactive team training.

Most of the early research in the area of team training focused on command and control (C2) systems to improve battle control procedures. Generally, the idea of training the skills of the team member for the benefit of the team was not considered. The first example of such training was the SIMNET project in which a number of tank simulators were interconnected to provide a collective C2 training environment [**2**].

The late 1990's have shown a substantial increase in research aimed at identifying team skills and developing techniques for training those skills in the team environment. However, the research still tends to concentrate on training basic procedural and psychomotor skills. Adaptability through improved decision-making along with procedural development has eluded the research focus. The VCGDT is offered as step in the direction of conducting this type of research.

2. What is Virtual Training

Virtual training simulations are those in which the trainee is immersed in a virtual world where physical actions such as driving a vehicle or firing a weapon have a direct visible effect on the synthetic world they are in. Constructive simulations are widely known as wargames. Tactical and strategic decisions are reflected in the movement of military icons on a map, testing the commander and staff’s ability to use their forces effectively. Live

simulations are the application of real equipment in mock combat scenarios or firing ranges. These allow pilots, tank drivers, and other soldiers to practice the physical activities of war with their real equipment [**1**].[1] The idea of training in a virtual world is appealing because it alleviates constraints posed by real world training. For one, the virtual training environments can significantly accelerate proficiency by exposing trainees to the kind of situations they are likely to encounter in the real world, but which could be hazardous or very expensive to practice in actual operational settings [**2**].[2] Budgetary constraints and public relations goals are relatively unaffected by virtual training.

Virtual training can also be controlled to an extent not possible in a real world environment. The characteristics of the training scenarios, situational cues, and decision outcomes can be provided as aids in the development of situational awareness, pattern recognition, and template building. Additionally, scenarios can be stopped, restarted, and manipulated at will. The degree with which control can be exhibited can lead to more effective training.

Finally, virtual training allows for the reinforcement of training objectives free of unwanted overhead and complexities of real-world training. Virtual training can be done quickly with highly adaptable situations with relatively low levels of preparation.

---

[1] Smith, Roger D., "Essential Techniques for Military Modeling & Simulation," *Proceedings of the 1998 Winter Simulation Conference*. Orlando, Florida.

[2] Bell, Herbert H., "The Effectiveness of Distributed Mission Training," *Communications of the ACM*, September 1999, Vol.42, No.9.

3. Present PC-based Systems

The shareware game Doom was probably the first commercial video game used for team training in the military. The U.S. Marine Corps Modeling and Simulation Management Office took the game and turned it into an urban combat scenario where opposition forces replaced nonhuman characters. Marine Doom taught its users a variety of team combat skills including attack sequencing, conserving ammunition, and observing the chain of command.

Other commercial games have been similarly altered for this purpose. The Army commissioned NovaLogic Inc. to include in its Delta Force 2 game features found in the Army's Land Warrior. Worn by infantry soldiers, Land Warrior is a complex, integrated system that includes a computer and radio, a GPS receiver, a helmet-mounted LCD display, and a modular weapons system that adds thermal and video sights and laser ranging to the standard M4 carbine or M16A2 rifle. The Army is currently evaluating the game's effectiveness compared to other training methods [**3**].[3]

A version of Bohemia Interactive's Operation Flashpoint is currently being adapted by Coalescent Technologies for use in their Deployable Virtual Training Environment (DVTE). The laptop-based system is designed to provide mission rehearsal and refresher training for Marine air support, light armored vehicles, infantry, and forward observers [**4**]. It's unclear how effective this system will be as a training aid aboard ship.

[3] Macedonia, Michael, "Games Soldiers Play," *IEEE Spectrum*, March 2002.

Numerous other infantry-based applications abound including many CQB oriented products like Rogue Spear and Ghost Recon, produced by Red Storm entertainment; and SWAT 3 made by Sierra. These commercial applications have many of the desired characteristics inherent in effective virtual environments, including state of the art graphical presentation and sound. However, these products are designed for the maximization of entertainment value not for military training. Although modifications to these applications can make them more suited for this purpose, its unrealistic to think that the underlying nature of the products can be adapted to fit an effective training model.

4. Effectiveness of PC-based Virtual Training

The discussion of PC-based virtual training effectiveness must first begin with the discussion of why this type of training is sought. Training in a virtual environment strives to emulate training in a physical environment for the very good reason that in most cases the physical one is better. Unfortunately, military organizations are forced to contend with a number of significant training constraints that make it difficult to train in the physical world with desired frequency. It's these constraints that force us to decide what limitations in virtual training we are willing to accept to improve that level of frequency.

Although increased frequency of training says nothing about its effectiveness, it's this concept that allows us to examine virtual environments as a solution. Now we are able to study PC-based virtual training environments in the right context to better identify the constraints associated with their use. A cost versus benefit analysis can then be

performed to determine if the virtual training solution is truly effective for the specific training needs.

Research has been performed in the analysis of PC-based virtual environments to determine effectiveness, although much of that research has focused on the older, more established applications such as flight simulators. One such example of this is an extensive study of the effectiveness of Microsoft's Flight Simulator by the U.S. Navy. They found that students who used such micro-simulation products during early flight training were 54 percent more likely to receive above-average scores in real flight tests than their peers who never used the software [**3**].[4]

Since similar research in the area of dismounted infantry, or CQB applications, is rare it's difficult to say that the same successes would apply. To develop what can be characterized as an effective PC-based virtual CQB training application we must further bound the problem. By bounding the problem we can more easily compare the attributes of a trainer with the attributes of effective training. The VCGDT bounds the problem by limiting itself to cognitive training alone. VCGDT was developed to focus on training effective decision making skills in a team environment. This allows the system's implementation to focus on the factors that make up effective decision training and those which are consistent with an effective virtual environment.

[4] Macedonia, Michael, "Games Soldiers Play," *IEEE Spectrum*, March 2002.

5. Attributes of Effective Decision-Making

A central theme emerging from an analysis of training processes is that decision-making stems from situational awareness and assessment, prioritization in dynamic task environments, and action/feedback structures in event management. Individuals operating collectively in mission-critical, task-oriented environments are, in essence, decision makers who together determine the final outcome of a mission [**2**].[5]

By determining what makes an individual an effective decision maker, you are then able to identify which knowledge, skills, and processes must be learned in order to achieve targeted performance. Table 1 presents Herbert H. Bell's list of attributes associated with good decision-making in a team environment. Table 2 shows Bell's areas of training focus to achieve these attributes.

Attributes of effective performers in team contexts
Flexibility - Ability to cope with environments that are ambiguous, rapidly changing, and complex.
Speed - Ability to make rapid decisions, often in the face of severe consequences.
Resilience - Ability to operate ambiguous, uncertain, stressful, and high-stakes environments without suffering degradations in performance.
Adaptability - Ability to recognize when and how to apply an action strategy and when to change or modify the strategy with problem demands.
Risk Management - Ability to quickly assess the risks in various courses of action, weigh the consequences, and payoffs.
Accuracy - Ability to state accurately what is obvious; quick and deep understanding and effective communication.

Table 1. Attributes of effective performers [**2**]

[5] Bell, Herbert H., "The Effectiveness of Distributed Mission Training," *Communications of the ACM*, September 1999, Vol.42, No.9.

Major skill areas in effective team training
Organized knowledge structures - Knowledge templates, relationships, and triggers.
Situation assessment skills - Make rapid, accurate assessments of situations; cue/pattern recognition and assessment of their significance.
Metacognitive skills - Select strategies, modulate strategies as problems unfold, engage in effective resource management, self-assess and adjust as necessary.
Reasoning skills - Analogical and casual reasoning, creative problem solving.
Domain-specific problem solving skills - Integration of domain knowledge with the other skills leading to rapid problem solving.
Mental simulation skills - Know when to simulate a scenario mentally and use it to evaluate strategies for novel problems.
Communication skills - Brevity, clarity, and timeliness of communication — both one-on-one and groups.
Other psychomotor skills - Specific to a training scenario.

Table 2. Major skill areas [**2**]

Bell suggests that these capabilities are crucial in achieving the ultimate objective of a desired level of expertise in both individual tasks and team missions. By placing individuals into a real-time virtual environment focused on team tasks, you are enforcing the very attributes associated with effective team performers in the real world. In addition, you're able to overcome the limitations of time and space by operating collectively while physically separated. This added benefit circumvents traditional real-world constraints.

B. VCGDT TRAINING FOCUS

The focus of the VCGDT design is a virtual training system that approximates a real world training facility to the extent that it enables team members to interact in a real time environment. Additionally it should be designed to enforce or learn cognitive behaviors associated with CQB

related missions. The training associated with these missions in the Marine Corps is conducted by the Special Operation Training Group (SOTG) located at both the I and II Marine Expeditionary Force (MEF).

1. Marine SOTG

The mission of SOTG is to provide training in special operations and warfare in diverse environments for the Marine Expeditionary Force. Specifically, they are tasked as being the sole entity within the Marine Corps to conduct special operations training, exercises, and evaluation in support of the Marine Expeditionary Unit (Special Operations Capable)(MEU(SOC)) training program. It is the CQB portion of this mission that VCGDT is designed to augment therefore its SOTG's doctrine that being emulated in its implementation.

2. MEU(SOC) Deployable System

The training gap left by the forward deployed environment faced by CQB trained Marines is one that the VCGDT is designed to help alleviate. Since the user of this system is primarily shipborne, an obvious constraint faced in VCGDTs development was the ability to operated effectively aboard a Navy ship. Chapter III discusses the details behind the software and hardware approaches to overcome the constraints discussed briefly here.

a. *Size Constraint*

The greatest problem associated with any ship-borne equipment is space available. Typically, Marines charged with maintaining CQB skills aboard ship are located on the largest of a three-ship Amphibious Readiness Group (ARG). The newest version of this ship is the Wasp Class

Landing Helicopter Dockship (LHD) (Figure 1). Although the larger of the ships, the LHD is still faced with significant space issues that make training systems with a large space signature unfeasible.

Figure 1. USS Bonhomme Richard (LHD 6).

To accommodate this constraint, the VCGDT is designed to the specifications of an individual Marine's requirement for standing space. The hardware associated with the system is worn, carried, or attached to a single laptop computer requiring only a Local Area Network (LAN) connection to obtain connectivity with other, similar, systems actively participating in the training. The idea behind this design is that it allows a team of Marines to train in individual small spaces around a single ship, or on different ships, without ever needing space for them to be physically located together. The physical separation on the ship is completely overcome through participation in the virtual training environment of the VCGDT.

An additional consideration is space overhead. Tracking systems that utilize light and sound many times have associated with them an overhead receiver, which is affixed to the ceiling or some equivalent. This type of system was inappropriate due to the lack of overhead space and infeasibility of installing such a system in what is essentially a community space.

b. *Bandwidth Constraint*

An issue faced by ship LAN administrators is bandwidth availability to support ship system, communication, and Internet service requirements. The VCGDT's software network design considers this constraint by limiting network traffic required for connectivity in the environment. The VCGDT server acts as the global network synchronizer for the environment. Client packets containing basic movement and action data are passed to the server. The server then periodically passes that information to the other clients to create a real time synchronized virtual world. Network packet sizes have been streamlined so that only the most basic synchronization information is required to pass over the network to any given client.

c. *Ship Construction*

The construction of the ship itself imposed an additional constraint for the VCGDT's tracking. Since the ship is composed primarily of metal, no tracking systems could be used that would be affected by being surrounded by such a composition. An electromagnetic tracking system is an example of this. The presence of any magnetic materials or power sources within or near the working area can

severely degrade its performance [**5**]. The VCGDT tracking was therefore limited to the use of inertial systems that would not be inhibited by either space or ship composition.

d. *Ship Motion*

The problem with ship motion is one that is not entirely dealt with in the design of the VCGDT. The system does eliminate some hardware implementation ideas because of its presence, however, its unknown the extent to which ship motion affects the operator's ability to effectively operate the system or protect himself from virtual sickness. This problem is left for further research.

III. APPROACH

The VCGDT's design is built around the procedural requirements laid out in the SOTG CQB training course offered to Marines assigned the MEU(SOC) prior to deployment. It's these requirements that were the foundation for VCGDT because of the impact each of them had on the ability of these Marines to make sound, doctrinal decisions. The VCGDT's focus is training these decision-making skills because it is intended for experienced users who can rely on rope memory for physical tasks. A task analysis, done in conjunction with this research, offers a breakdown of that decision process [**6**]. Since the VCGDT's design is incapable of implementing some of these requirements based on shipborne physical constraints, their discussion includes suggestions as to why they were omitted.

A. PHYSICAL INTERACTION

Physical interaction in a CQB environment provides an operator a method of non-verbal communication that is essential to effective incorporation of accountability, maneuver, and security procedures. The necessity of this communication medium is apparent in environments where excess noise, darkness, or both, limit the operator’s ability to interact with fellow operators using voice and radio communications alone.

A virtual training environment is one that does not easily lend itself toward incorporating these types of interaction. One primary benefit of the VCGDT's design is that a team is able to operate together in a virtual

environment even when physically separated from each other. This, of course, makes it infeasible with today's technology to include the ability to physically interact with one another. The loss of this important aspect of CQB operations is an omission made as a trade-off to accommodate the aforementioned benefit.

The following is a list (not all-encompassing) of physical interactions within the CQB environment. The list is presented as a suggestion for follow-on research into this added benefit.

1. Leg Bump

When encountering a decision-point that requires multiple shooters to act together, a leg bump is used to initiate the action. This gesture occurs when the number two man in the stack, or line of operators, recognizes that a need to act exists. He closes his distance behind the number one man, quickly and firmly bumps the number one man's hamstring using his knee, and announces, "with you!" This gesture lets the number one man know that he can now proceed with the action with the confidence that the number two man will follow him.

The purpose of the leg-bump is two-fold: One, the number one man doesn't have to look back to know he's being supported, and two, despite a dark/noisy environment, the number two man is still able to articulate his intent.

2. Turn and Go

Following the successful clearing of a room, the CQB team must ensure that security is at all times maintained. They must maintain a physical presence in the room to ensure that no hostile repopulation of the room occurs.

During the consolidation and exfiltration process it is necessary for the team to systematically step back security in such a manner that allows for security of the occupied spaces to be maintained. The process is referred to as, "peeling back".

Peeling back is done through team leader direction at the maneuver level but done using touch at the individual operator level. Since it is crucial that an operator maintaining security not remove his attention from the secured area, he is forced to wait until someone places his hand upon his shoulder telling him to "turn and go". Once this is done, he has no doubt that he is the one being spoken to. He immediately turns and abandons his security position for the designated consolidation area.

3. HUTS Report

Following consolidation of the operators in the consolidation area, it is the Platoon Sergeant's responsibility to generate the appropriate information for the Team Commander to produce the Hostages, Unidentified, Terrorists, and Shooters (HUTS) report. In each category of the report, the Commander will tell higher headquarters the number of individuals in each of the aforementioned categories. Security and medical personnel determine the number of hostile and unidentified individuals, and downed shooters. The number of healthy shooters and dead terrorists are determined by the Platoon Sergeant using two methods of physical communication: Touching a shooter down and grabbing a shooter's hand to indicate a terrorist count.

To ensure accountability, the Platoon Sergeant counts each healthy operator in the consolidation area. He does

this by touching each one on the head. When an operator is touched, he promptly kneels to ensure that he is counted only once. Once all of the operators are kneeling, he knows he has counted each operator in the consolidation area one time.

Once they are all kneeling, the operators are asked to raise their hands and present the number of terrorists they have personally dead-checked using their fingers. The Platoon Sergeant walks from man to man grabbing each of their raised hands, totaling the number of terrorists. Once an operator's hand is grabbed, he immediately lowers it to avoid duplication in the count.

Each of the aforementioned methods is designed to allow for accurate numerical information to be articulated from the operators to the Platoon Sergeant in both dark and noisy environments.

B. NON-VERBAL COMMUNICATIONS

Non-verbal communication is essential to the CQB environment as a secondary means of articulating important queues and information. Under normal, non-surreptitious conditions voice communications are used as the primary means of communication; however, they are always used in concert with non-verbal to alleviate confusion and possible misunderstanding. One example, of many, includes the dead checking of downed hostiles. An operator approaches a downed hostile under the security of another operator. After clearing any weapons from the body, he "thumps" the hostile's eyeball to ensure that he is dead. Once successful, the operator yells, "down!" while simultaneously holding up his hand in a "thumbs down" position.

C. EFFECT OF STRESS

Stress is one of the main contributors to the problem of job performance in the CQB context. The effects of severe stress degrade all decision-making skills, motor functions, and communication ability. Combating these effects may be as simple as incorporation of stressors into the training environment. An idea posed by LtCol David Grossman suggests that inoculation of stress can occur through consistent exposure. He points out that this is exactly what occurs in boot camps, where recruits are faced with seemingly sadistic abuse and hardship [**7**].[6] Incorporation of these types of inoculating factors could prove beneficial to combating the problem of stress-induced decision degradation.

D. TRAINING FACILITY MODELS

In the pursuit of appropriate CQB training environments, it is essential that an individual undergoing the training be exposed to a virtual environment that best tests/trains his ability. In some contexts, the logical starting point in searching for an environment to model would probably be to look at the real-world application of the activity being tested/trained and attempt to model it. Unfortunately, the act of engaging in battle in close quarters does not occur in any one particular scenario. In fact, the spectrum of different situations one might encounter cannot be reasonably modeled.

This problem is one that has already been provided a solution in the real-world training of Marines tasked with this mission. The Marine Corps has taken the approach of

[6] Grossman, Dave, Lt.Col., "On Killing," Little, Brown and Company, 1996.

building scenarios that test skills over adaptability. The assumption is that by training an individual well enough and with enough solid repetition, his reaction will be the same regardless of the situation.

Following the Marine Corps ideology in CQB training is the best solution to model selection. Cognitive decision-making ability has an infinite number of possible variations. By focusing on standardization and meaningful repetition, an individual's decision-making ability can mirror the performance of his physical skills. To achieve this, the VCGDT uses the facility in which the Marines themselves have been trained. By standardizing the environment in which the decisions are made, a more formal baseline exists for testing the standardized decisions themselves.

E. DECISION NOISE

1. Problem With Noise

One problem with testing the ability to make decisions in a virtual environment is simulating the actual conditions that occur simultaneous to the decision. The aforementioned topic discusses the model that the individual will be subjected to. Noise, however, is an independent issue.

Noise represents all exterior stimuli an individual encounters in a typical CQB environment. Noise can mean acoustic interference, stress, fear, visual impairment, or anything that distracts the individual from making sound, timely decisions. A virtual environment that fails to include these distractions is referred to commonly as a

"canned scenario". A canned scenario is one that favors simplicity of instruction over realism.

As an example, a Marine faced with an improvised explosive device (IED) in a room of a building will need to make a vital decision about that device. However, before, during and after discovering the device, the Marine has a list of concerns and emotions that he is contending with while entering that room. He is thinking about preserving his life and the lives of his fellow Marines. He is thinking about his position and listening to the verbal queues of others. He is battling the fatigue of wearing his equipment and holding his weapon to his shoulder. He's trying to focus on the threat before him while drowning out the sound of machine-gun fire and flash-bangs. These and other factors contribute to his ability to make a sound decision.

2. VCGDT Noise Design

The VCGDT's design is based on this premise. Although it's difficult to incorporate much of the noise typically associate with a CQB environment, many are effectively reproduced. Marksmanship, for example, is a method of drawing focus from the operator in a virtual world. While the VCGDT does not promote marksmanship training through its use, it does require the individual operator to conduct basic aim and shoot movements.

The VCGDT's instructor avatar is another implementation for this purpose. Introduction of a visible authority induces operator stress and encourages a distracted environment. The concept attempts to mirror the stress the training environment the operator is familiar with.

F. TRAINING SUPERVISION

Supervision of training is critical to the process of evaluation. Keeping statistics and monitoring correct/incorrect steps in the process of making a standardized decision can only serve as a superficial measure of effectiveness. Observation by a human is therefore essential to the testing and training of other humans.

Introducing a neutral third party into CQB training is not a new concept. In real-world CQB training, Marines are very accustomed to being constantly monitored by their peers, their instructors and their superiors. The process exists as a way to monitor success, ensure safety, increase the learning curve for on-looking peers, and a long list of others functions. However, once the training moves from its normal cycle to being done closer to and during the deployment phase, it becomes critical in the readiness evaluation of the unit.

In the VCGDT prototype, the instructor avatar is visually distinct having a different color uniform and being unarmed. The instructor is able to communicate with local operators using spatialized communications or to the group using broadcast communications. As an instructor in the environment, you have the unique ability to perform non-verbal communications using hand and arm gestures. Additionally, you have the ability to view the environment through either the avatar's viewpoint or through a top-down position.

G. COMMUNICATIONS

Communications is the key to a successful mission. It can take the form of radio communications with team leaders or higher, it can be of a direct verbal nature local to the room, and it can be of a non-verbal nature.

There are many echelons of situational awareness that are needed in a close quarter battle space. An individual must have a rich understanding of things in close proximity to himself and those that are occurring in other areas of a building or ship.

1. Non-local Communication

Radio communications help the operator do this for areas he cannot observe. Normally he is equipped with a "throat mike" plugged into a VHF/UHF encrypted saber radio attached to his back. Conversation over the system requires the user to depress a small button attached to his chest. Incoming communication is heard over a small earpiece placed in the left ear. While the individual conducts his piece of the mission, he is constantly listening to updates being transmitted through his earpiece.

2. Local Communication

For areas the operator can observe, he is listening to verbal and non-verbal communication from his teammates. Hand signals are common and necessary to room clearing because the excessive noise makes it difficult to hear what those near you are saying.

The VCGDT takes both the verbal and nonverbal considerations into account. It's basic design allows both spatial and broadcast communications. The current hardware

configuration does not allow for the use of actual radios, however, they can easily be incorporated as part of the voice over IP (VOIP) network communications system. Although the VCGDT does not support unique non-verbal communications between operators, it does allow an instructor avatar the use of a set of hand and arm signals to communicate non-verbally or augment verbal commands.

IV. IMPLEMENTATION

A. SOFTWARE

1. Overview

The VCGDT prototype was developed using C++ classes as an implementation of the application programming interface (API) GFLIB. The GFLIB architecture has been developed over the past year through a cooperative effort of several students overseen by Erik Johnson. GFLIB is modeled after MultiGen Paradigm's VEGA development library, but is composed primarily of three other APIs: Gizmo3D is the basic game engine and major part of the unexposed API; Microsoft DirectX 8.0 provides audio, voice, networking and input device support; and Sglib provides basic mathematical computation support. The underlying components of these APIs are hidden from the developer in GFLIB method calls. GFLIB uses a VEGA-like naming convention: all methods are pre-pended with the letters 'gf' indicating a library method call or access. GFLIB follows the same set of construction rules, as does VEGA. The rules govern the relationships between GFLIB entities. For instance, you may have an object you wish to render. To do so you would need to create a scene (to represent a scene graph), then add the object to it [**8**].

The VCGDT implementation uses GFLIB library to construct an application specific library called *gfAvatar*. This library consists of the functionality needed to build and manipulate virtual representations of physical participants. *gfAvatar* uses the commercial API DI-Guy 4, created by Boston Dynamics Inc. (BDI), to produce the

avatars and associated animations. Method calls from BDI are embedded in the framework of the *gfAvatar* classes making it specific to this commercial product.

2. Avatars

The VCGDT allows for the user to virtually represent himself as two different avatars, a shooter or an instructor. Avatar selection affords the user different attributes in the environment.

Figure 2. Shooter Representation.

a. *Shooter*

As a shooter, a user's avatar is represented in green camouflage utilities and carrying a rifle (Figure 2). The character itself is the Soldier 2 Boston Dynamics

design. The avatar's capabilities roughly match those of a real-world operator; having the ability to maneuver through a building (Figure 3), aim and fire his weapon (Figure 4), and communicate with other participants in the environment using spatialized and global voice communications.

Figure 3. Walking at Low-Ready.

Figure 4. Aiming Weapon.

b. *Instructor*

An instructor is represented as an unarmed Boston Dynamics Soldier 2 with desert-style camouflage utilities. Apart from being visually distinct, the instructor has the same movement and voice communication characteristics as a shooter. The instructor does, however, have some additional features than that of a shooter. First, he has the ability to view the environment from an overhead perspective (Figure 5). He can collectively view all of the other participants and space from above to allow for improved situational awareness in instructing shooters.

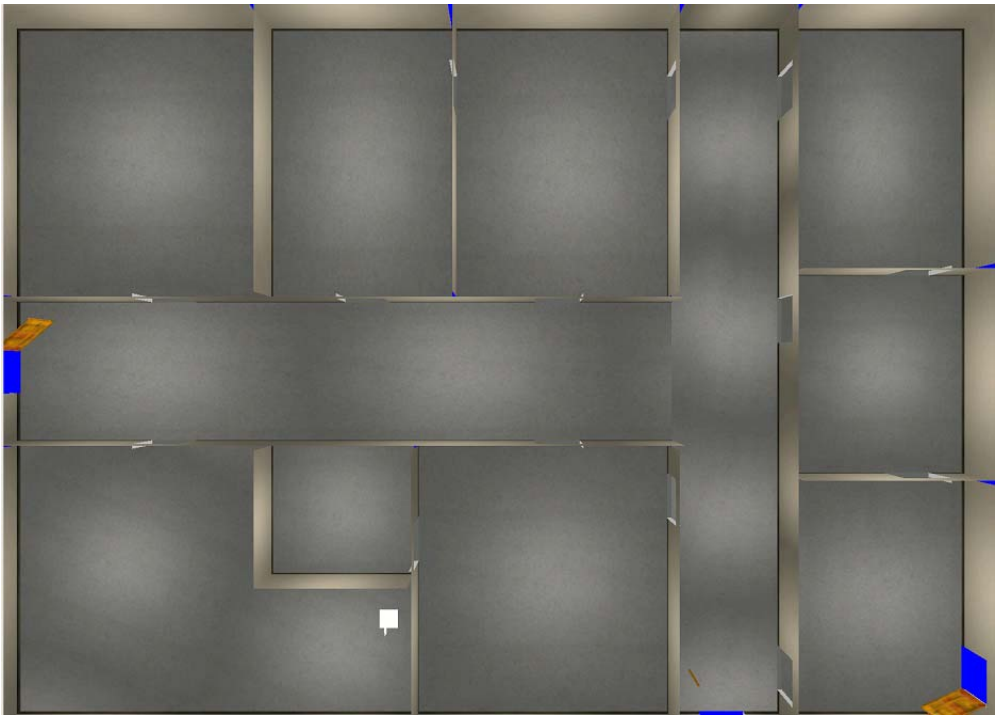

Figure 5. Instructor's Overhead View.

Additionally, an effort was made to incorporate non-verbal communication into the instructor's character because it is such an important part of real-world communication. He is able to manipulate his avatar to make a series of hand and arm gestures including the ability to:

point to the left and right +90/-90 degrees of the avatar's heading; point directly forward along the avatars heading; motion to stop by raising the right hand in the air with elbow positioned at a 90 degree angle from the avatar's side; motion to back away by pushing the hand forward and back from the stop position; and motioning to come forward using both hands moving back into the avatar's chest.

3. Implementation Details

This section briefly describes the *gfAvatar* library and each of its inner classes. Each of the sub-sections describes a single class, starting with a general description of the class, and followed by a table of the class's data members and it's most important functions.

a. *gfAvatarDataset Class*

The *gfAvatarDataset* class defines the production of a *gfDataset* which is the mechanism used to load data into the scene graph. The *gfDataset* class calls a *LoadFile* function to load in geometry from disk in an OpenFlight format. The *gfAvatarDataset* can be added to a *gfAvatarObject* that can be subsequently added to the gfScene. The *gfAvatarDataset* data members are listed in the table below.

Name	Type	Description
mNode	*gzRefPointer <gfAvatarNode>*	Stores the gfAvatarNode created by the constructor

Table 3. Data members of *gfAvatarDataset* class

The table below lists some of the functions provided by the *gfAvatarDataset* class.

Name	Return	Description
GetGZNode(void)	*gzNode**	Returns mNode

Table 4. Functions of *gfAvatarDataset* class

b. *gfAvatarMotion Class*

The *gfAvatarMotion* class is a motion model designed to approximate the characteristics of human motion for an instructor. It is derived from the pure virtual class *gfMotion*. This class can be used to position *gfAvatarPlayer* and *gfObserver* objects. A *gfInput* object is attached to it to provide user-based input from various devices. The *gfAvatarMotion* data members are listed in the table below.

Name	Type	Description
mSideVel	*float*	Stores value for sidestep through keyboard input
mFwdVel	*float*	Stores value for frontstep through keyboard input
eyepoint	*gzRefPointer<gfObserver>*	Pointer to observer
scene	*gzRefPointer<gfScene>*	Pointer to scene.
mAction	*gfAtlasActionEnum*	Stores enumeration for avatar action.
walkDirection	*sgVec3*	Stores walk vector for collision

		detection.
point	*bool*	True if avatar is pointing.
pose_array	*float**	Stores array of values to describe avatar pose.
weights_array	*float**	Stores array of values to describe avatar custom movement.
po_handle	*diguyPoseOverrideHandle*	Stores a pose.
di_handle	*diguyCharacterHandle*	Stores diguy character.
isPointing	*bool*	True if avatar is in the act of pointing.

Table 5. Data members of *gfAvatarMotion* class

The table below lists some of the functions provided by the *gfAvatarMotion* class.

Name	Return	Description
ProcessInput()	*void*	Reads input devices.
CreateNewWalk()	*void*	Adds gfAvatarMotion to list of motion models.
SetAvatarAction(gfAtlasActionEnum)	*void*	Sets mAction.
GetAvatarAction()	*gfAtlasActionEnum*	Gets mAction.

DetectCollision(gfPosition)*	*gzBool*	Returns true if position is too close to object in path.
SetDirection(float, float, float)	*void*	Sets sgVec3 based on x,y,z position.
GetDirection(sgVec3)	*void*	Gets direction of avatar travel.
GetForwardVelocity()	*float*	Returns mFwdVel.
ClassID()	*const int*	Returns GFTYPE_AVATAR_MOTION denoting the type of motion model used.
PoseOverride(diguyCharacterHandle)	*void*	Implements the avatar point pose.
IsPointing()	*bool*	Returns true if avatar is pointing.

Table 6. Functions of *gfAvatarMotion* class

c. *gfAvatarNode Class*

The *gfAvatarNode* class is instantiated by the *gfAvatarDataset* constructor. A *gfAvatarNode* is created as a child of the *gzNode* class of the Gizmo 3D library. Its purpose is to define and monitor the interaction between BDI and the scene graph. It establishes the graphical parameters for the BDI geometry including texture and BDI feature information. The *gfAvatarNode* data members are listed in the table below.

Name	Type	Description
mCharacterHandle	*diguyCharacterHandle*	Handle to BDI avatar.
mTimer	*ulClock*	Stores ulClock system timer.
avatarType	*char[]*	Stores file name for avatar.

Table 7. Data members of *gfAvatarNode* class

The table below lists some of the functions provided by the *gfAvatarNode* class.

Name	Return	Description
GetCharacterHandle()	*diguyCharacterHandle*	Returns handle to avatar.
GetAvatarType()	*char**	Returns avatarType[]

Table 8. Functions of *gfAvatarNode* class

d. *gfAvatarObject Class*

The *gfAvatarObject* class is used to initialize the state information of the avatar created by the *gfAvatarNode* that's passed to the *gfAvatarObject* through the *gfAvatarDataset*. It provides the developer with a handle to the *gzNode* information so that it can be manipulated in the scene graph. The *gfAvatarObject* data members are listed in the table below.

Name	Type	Description
mCharacterHandle	*diguyCharacterHandle*	Handle to BDI avatar.
eyepoint	*gzRefPointer<gfObserver>*	Handle to gfObserver.

walk	*gzRefPointer<gfMotionWalk>*	Handle to motion model.
fireWeapon	*bool*	True if weapon is being fired.
shotComplete	*gzBool*	True if gun shot is over.
avatarType	*char[]*	Stores file name for avatar.

Table 9. Data members of *gfAvatarObject* class

The table below lists some of the functions provided by the *gfAvatarObject* class.

Name	Return	Description
SetShooterAction (gfActionAvatarEnum)	*void*	Matches Enum to BDI Enum list for shooter action.
SetInstructorAction (gfAtlasActionEnum)	*void*	Matches Enum to BDI Enum list for instructor action.
SetFiring(gzBool)	*void*	If parameter true, sets gun to fire mode.
SetAiming()	*void*	Aims weapon using current heading and pitch.

SetGunPitch (diguyCharacterHandle, float, float)	*void*	Sets heading and pitch of gun.
GetDIHandle()	*diguyCharacterHandle*	Returns handle to BDI avatar.
GetAvatarType()	*char**	Returns avatarType[].

Table 10. Functions of *gfAvatarObject* class

e. *gfAvatarPlayer Class*

The *gfAvatarPlayer* class defines the production of a *gfPlayer*. The *gfAvatarPlayer* contains *gfAvatarObjects* and a motion model. It can be positioned either using the assigned motion model or by specifying the new position directly. The objects that are attached to the player will be moved based on the player's position. The player contains all of the functionality that is specific to the objects associated with it. The *gfAvatarObjects* associated with the player must be put in a *gfScene* in order for them to be rendered. The *gfAvatarPlayer* data members are listed in the table below.

Name	Type	Description
hitLocation	*gzVec3*	Stores XYZ location where gunshot hits object.
locNormal	*gzVec3*	Stores the normal associated with the hitLocation.

dsLocked	*gzBool*	Flag used to control gunshots.
scene	*gzRefPointer<gfScene>*	Handle to gfScene.
bulletPos	*gzRefPointer<gfPosition>*	Stores XYZ of bullethole.
eyepoint	*gzRefPointer<gfObserver>*	Handle to gfObserver.
maskResult	*gzULong*	Stores mask of object at hitLocation.

Table 11. Data members of *gfAvatarPlayer* class

The table below lists some of the functions provided by the *gfAvatarPlayer* class.

Name	Return	Description
SetHitLocation (gzDynamicArray <gzIntersectorResult>)	*void*	Sets hitLocation.
SetHitNormal (gzDynamicArray <gzIntersectorResult>)	*void*	Sets locNormal.
KillAvatar (gfMotionWalk, const char*, const char*)*	*void*	Instructs avatar to assume dead action.
GetHitLocation()	*gzVec3*	Returns hitLocation.
GetHitNormal()	*gzVec3*	Returns locNormal.

ProcessHit(gzVec3, gzVec3)	*void*	Assigns bullet hole position based on hitLocation and locNormal.
DetectHit (gfPosition, sgVec3)*	*gzBool*	Returns true if Isector from gun encounters object in scene.
GetIsDSLocked()	*gzBool*	Returns dsLocked.
MotionModel()	*gzBool*	Returns true if motion model assigned to object.
GetKillObjectFromMask (gzLongLong)	*gfObject**	Returns object associated with mask.
GetMotionFromMask(gzLongLong)	*gfMotionWalk**	Returns motion model associated with mask.
GetPlayerMask(gzNode)*	*gzLongLong*	Returns mask associated with gzNode.
GetPlayerFromMask (gzLongLong)	*gfAvatarPlayer**	Returns player associated with mask.

Table 12. Functions of *gfAvatarPlayer* class

f. *gfMotionWalk Class*

The *gfMotionWalk* class is a motion model designed to approximate the characteristics of human motion for a

shooter. It is derived from the pure virtual class *gfMotion*. This class can be used to position *gfAvatarPlayer* and *gfObserver* objects. A *gfInput* object is attached to it to provide user-based input from various devices. The *gfMotionWalk* data members are listed in the table below.

Name	Type	Description
mSideVel	*float*	Stores value for sidestep through keyboard input
mFwdVel	*float*	Stores value for frontstep through keyboard input
eyepoint	*gzRefPointer<gfObserver>*	Pointer to observer
scene	*gzRefPointer<gfScene>*	Pointer to scene.
mAction	*gfActionAvatarEnum*	Stores enumeration for avatar action.
walkDirection	*sgVec3*	Stores walk vector for collision detection.
isAiming	*gzBool*	True if aiming
isFiring	*gzBool*	True if firing
isDead	*gzBool*	True if action is dead.
fireWeapon	*gzBool*	True if gun has fired.

shotComplete	*gzBool*	True if shot successfully taken.
gunHPR	*sgVec3*	Store HPR of gun object.
gun	*gzRefPointer<gfObject>*	Handle to gun gfObject.
flash	*gzRefPointer <gfObject>*	Handle to gun flash gfObject.

Table 13. Data members of *gfMotionWalk* class

The table below lists some of the functions provided by the *gfMotionWalk* class.

Name	Return	Description
ProcessInput()	*void*	Reads input devices.
CreateNewWalk()	*void*	Adds gfMotionWalk to list of motion models.
SetAvatarAction (gfActionAvatarEnum)	*void*	Sets mAction.
GetAvatarAction()	*gfActionAvatarEnum*	Gets mAction.
DetectCollision(gfPosition)*	*gzBool*	Returns true if position is too close to object in path.
SetDirection (float, float, float)	*void*	Sets sgVec3 based on x,y,z position.

GetDirection(sgVec3)	*void*	Gets direction of avatar travel.
GetForwardVelocity()	*float*	Returns mFwdVel.
Position(gfPosition)*	*void*	Sets motion model position.
SetIsAiming(gzBool)	*void*	Sets isAiming.
SetIsFiring(gzBool)	*void*	Sets isFiring.
SetGunHPR(float, float, float)	*void*	Sets HPR of gun.
GetGunHPR(sgVec3)	*void*	Gets HPR of gun.
SetIsDead (gzBool,const char, const char*)*	*void*	Sets avatar action to dead if gzBool true.
GetIsAiming()	*gzBool*	Returns isAiming.
GetIsFiring()	*gzBool*	Returns isFiring.
GetIsDead()	*gzBool*	Returns isDead.

Table 14. Functions of *gfMotionWalk* class

g. *myNetwork Class*

The *myNetwork* class is derived from the *gfNetwork* class and is designed to package, transmit, and receive data updates and synchronization information over a network. It implements a series of virtual methods and

contains a series of inner classes used to package the data into categorical and transmittable form. The virtual functions implemented by the *myNetwork* class are listed in the table below.

Name	Return	Description
ReceiveMessage (gfMessageData)*	*void*	Receives network messages.
DestroyPlayer (gfDestroyPlayerData)*	*void*	Destroys a network player
CreatePlayer (gfCreatePlayerData)*	*void*	Creates a new network player
UserOnNotify (gzNotifyMessage)*	*void*	Notifications not handled by this class get passed to this method.
ConnectionComplete()	*void*	Called when local player's connection has completed.

Table 15. Virtual functions of *myNetwork* class

The table below lists some of the other functions provided by the *myNetwork* class.

Name	Return	Description
Shutdown()	*void*	Terminates player from network connection.
ProcessPlayerPosition (gfMessageData)*	*void*	Updates a remote player's local position

ProcessPlayerState (gfMessageData)*	*void*	Updates a remote player's local information
ProcessShooterAction (gfMessageData)*	*void*	Updates a remote shooter's local action.
ProcessInstructorAction (gfMessageData)*	*void*	Updates a remote instructor's local action.
ProcessBulletHole (gfMessageData)*	*void*	Updates local bullethole positions from remote position information.
ProcessAction (gfMessageData)*	*void*	Used for generic action assignment to non-player avatars
SendPlayerPosition (gfPosition)*	*void*	Sends local player position to the network.
SendPlayerState()	*void*	Sends out player's state to synchronize network.

Table 16. Functions of *myNetwork* class

The table below lists the inner classes of the *myNetwork* class.

Name	Parent Class	Description
gfPlayerStatePacket	*gfBasePacket*	Packages player XYZ HPR and avatar type.
gfInstructorActionPacket	*gfBasePacket*	Packages instructor action.
gfShooterActionPacket	*gfBasePacket*	Packages shooter action and gun HP.
gfBulletHolePacket	*gfBasePacket*	Packages bullet hole XYZ HPR.
gfActionPacket	*gfBasePacket*	Packages action, player ID and geometry ID.

Table 17. Inner classes of *myNetwork* class

B. HARDWARE

The hardware described in this section represents the input and display devices associated with a user of the VCGDT (Figure 6). There are obvious computational components that are vital to the hardware implementation; however, this document focuses on those devices specific to the application itself.

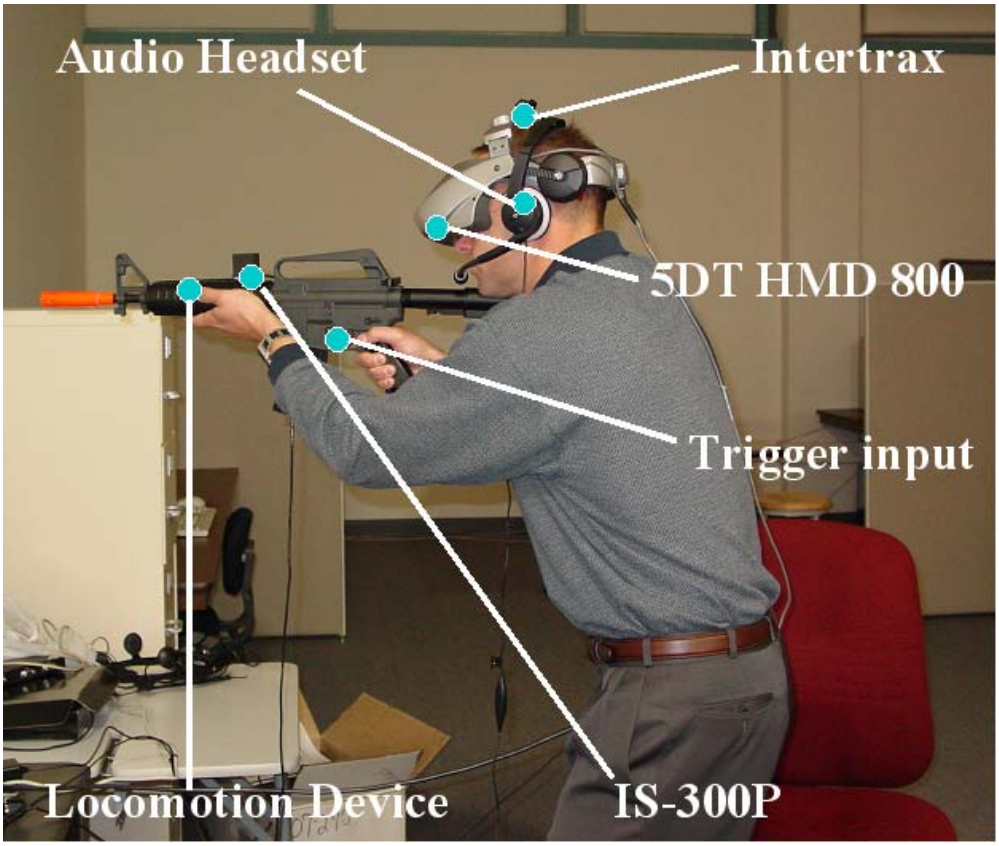

Figure 6. Hardware Components.

1. Motion

The definition of movement is a change in place or position. In the context of this document, the term movement is used to describe the changes in an avatar's direction, global coordinate position, or posture. Posture is used to describe the relative arrangement or placement of an avatar's body parts to assume some type of pose.

Movement inside the CQB environment is more sophisticated than that needed to maneuver ones' self in an ordinary context. Even on the open battlefield, slight differences in movement have little impact on an operators' ability to perform a task effectively. When you place that operator in a close quarters environment, control and freedom of movement become more essential to every aspect

of the mission. Because these nuances so greatly effect the operator, representing traditional real world movement in a virtual environment involves a much greater level of complexity than would be needed to represent movement in another context. To help alleviate this complexity, some generalities need to be made.

a. *Foot Control*

Foot control represents all of the data needed from the operator to determine the direction and manner in which the operator wishes to travel in the virtual environment. To simplify this problem, the assumption is being made that an operator requires to only step in four distinct directions: forward, backward, side-step left, and side-step right. Although this does not accurately represent real world movement, the omission of omni-directional foot control was designed to alleviate complication of system operation. By allowing omni-directional control, the potential for human error in the virtual control increases, which translates to a higher learning curve for system operation. While omni-directional movement is prevented, the four-direction control system can still be manipulated to simulate most directions on a 360-degree plane. This manipulation is done through the combination of step and direction controls.

Numerous types of devices were considered for this purpose. The following table describes products that were considered for reproduction.

Device	Description
Uniport	The user pedals to simulate walking or running. The metaphor is that of cycling rather than natural bipedal locomotion [**9**].
Treadport	A treadmill with the user being monitored and constrained from behind via a mechanical attachment to the user's waist [**9**].
Omni-Directional Foot Pedal	A disc that reads direction and pressure applied by a user on a 360 degree plane requiring him to physically changing his orientation in the real world to change his direction of travel [**10**].
Pressure Mat	A rubber mat with embedded sensors which sample the pressure exerted by a user as he performs various actions [**10**].
Virtual Motion Controller	A user stands on a concave disc. Movements from the center result in motion in the VE in the same direction as the movement on the disk, and with a velocity proportional to the distance from the center [**11**].
Cybershoe	Pressure and bend sensors embedded in a foam insole inserted between the sole of a shoe and the sneaker's insole. Wireless transmission of data [**12**].
Finger Sleeve	A pressure sensitive pad in a wearable finger sleeve [**10**].

Weapon-Mounted Joystick	An elastomer button on top of a four-direction pressure pad [**10**].

Table 18. Locomotion devices considered

The weapon-mounted joystick was chosen because of the simplicity and all-purpose nature of the device. Many of the aforementioned devices, like the pressure mat, seemed a better fit for the VCGDT prototype, however, time constraints made the use of these devices infeasible for use with this research. In the proof of concept stage, it was determined that the joystick would temporarily meet the minimal requirements.

The joystick's design stemmed from research conducted by the Southwest Research Institute (SWRI). Their implementation of a simple four-direction device was mimicked using pressure sensitive binary resistors taken from a Logitech Wingman wireless game pad. They were mounted to the handguards of the weapon such that the resistors can be manipulated with the weak-hand thumb.

The decision to use the weapon-mounted joystick did leave a gap in what would traditionally be thought of as an ideal locomotion device. Research has shown that the introduction of vestibular cues in virtual locomotion can improve some aspects of movement. On one hand, its been demonstrated that vestibular cues in locomotion can improve a user's ability to form a mental map for better navigation. Conversely, this same research indicated that basic movement is better with the use of a joystick, omitting all vestibular cues associated with body movement [**11**]. All in all, the decision to omit the introduction of such sensory cues would probably be due to the fact that

their introduction, unless highly accurate, can lead to unpredicted and unwanted effects [**13**]. Determining which cues are appropriate for this environment is left for future work.

b. *Posture*

Posture in a close quarter environment can be an important tool in maintaining both safety and stability for the operator. Assuming a low profile during mission execution is key to decreasing the probability of becoming a quick target for a hostile. As such, it becomes a routine part of the CQB training process and an almost unconscious act by the operator.

Posture contributes to stability through the use of a low, tight stance with feet close together and facing in the direction of travel. Generally the operator will be taught to walk heel to toe in a smooth controlled fashion, coupled with the elbows pressed against the midsection to support the weapon. Together, this posture helps stabilize the operator's aim, even while moving.

The VCGDT's design was constrained with regard to posture. The Boston Dynamics avatars used in the prototype required a level of detailed customization that exceeded original performance thresholds. As a result, this aspect of operator movement was not modeled. Although left to future work, its need is important enough to mention here.

c. *Gestures*

The keyboard is used as the input device to drive the instructor's non-verbal gestures. Ideally the motions would be associated with actual hand and arm movements,

however, due to time and resource constraints, its left to future hardware implementation improvements.

2. Weapon

The system's weapon is a mock M4-style rifle that has been equipped with a resistor to capture input data from the trigger. The resistor comes from the same Logitech game pad device used in developing the aforementioned joystick. The heading and pitch of the weapon are controlled through the use of an Intersense IS300 Pro inertial tracker mounted to the top of the weapon.

The control of direction is a function of the heading of the weapon being used by the operator. Since the assumption is that the weapon will always face in the direction the operator wishes to move, the control of direction of step is mapped to that direction. By manipulating the direction of the weapon in combination with steps taken forward, backward, or to the side, a close approximation to omni-directional movement can be obtained. Both the Intersense tracker and a resistor affixed to the trigger record weapon manipulation. The inertial tracker gauges the pitch of the weapon so that the operator is able to raise and lower the virtual weapon through direct physical pitch manipulation of the physical one.

Additionally, the trigger resister digitally monitors the relative position of the physical weapon's trigger to determine whether or not it is depressed. This enables the operator to fire the virtual weapon in a manner consistent with its real world counterpart.

3. Head Mounted Display (HMD)

a. *Control*

Head movement is controlled through the use of an Intersense Intertrax inertial tracker. The tracker is affixed to the HMD and allows for 3 degrees of freedom to accommodate movement of the head independent of the orientation of the body or weapon. While the weapon-mounted tracker directly controls the *gfAvatarObject* that represents the avatar's weapon, the head-mounted tracker controls only the viewpoint in the virtual world. This simulates the ability to move one's head independent of the body, but at the same time prevents manipulation of any geometry by head movement.

The idea of head/body independence is key to enhancing the user's ability to quickly navigate while maintaining situation, or environmental, awareness. An experiment was performed at the Georgia Institute of Technology to compare and contrast this type of virtual steering to a gaze-controlled steer. The study used an HMD, or viewpoint, to act as the gaze-directed steering device. It showed that by using some element of the body to steer, users were able to more quickly and accurately move in an environment they used with at least some frequency. While using gaze-directed steering proved effective for absolute motion, it required too much head movement, distracting users from their task [**14**]. Since speed, accuracy and focus were all vital performance issues within the VCGDT, the more logical choice was to use the weapon to steer as an extension of the body.

b. *Display*

The three largest considerations for finding a hardware match for the users view of the virtual world were Field of View (FOV), resolution of display, and stereoscopy. Although current research in the use of stereoscopic presentations has failed to find a direct correlation between its use and increased presence, it has shown benefit. It's been demonstrated that 3D cues exist in a stereoscopic presentation that increase a user's ability to judge short distances [**15**]. When evaluating its use in a CQB application, this benefit seemed significant enough to include in the criteria for an HMD search. Since stereoscopy, FOV, and resolution seem to promote presence in the VCGDT, in absence of conclusive evidence, the argument over which is most important is omitted from this document.

Once these criteria were determined, the HMD 800 Series head mounted display, produced by Fifth Dimension Technologies was chosen as the best low-cost mix of these factors. The HMD 800 has a FOV of 28 degrees horizontal and 21 degrees vertical. Although these specifications fall short of many high-end systems, the HMD 800's combination of an 800 horizontal pixels by 600 vertical pixels resolution and stereoscopic view makes it the most balanced. The sound system associated with the system consists of Sennheiser HD 25 closed dynamic headphones that operate at a sound pressure level of 120 dB at 1KHz. Improvements to the sound requirements and associated hardware are left to future work.

c. *Cyber-Sickness*

It should be noted that although cybersickness is not fully addressed by the design of the VCGDT, it was a consideration in the HMD selection. Research has shown that there is a direct correlation between weight placed on a user's head and motion sickness incurred through the use of a virtual environment. Heavier HMDs increase the effective weight of the head, in turn, increasing the potential for motion sickness [**16**]. The HMD 800 weight is 594 grams, which is far below the threshold indicated by this research to induce motion sickness problems.

V. CONCLUSIONS

A. GENERAL DISCUSSION

This thesis has briefly discussed the technology of virtual training and simulation along with the VCGDT's focus with regard to this technology. It has detailed the important characteristics of a sound CQB environment and shown the prototype's implementation of those characteristics. Additionally, it discusses the constraints associated with such a system being used aboard ship and methods for accommodating these constraints in the hardware implementation.

Since the VCGDT is a prototype system, the following conclusions and future work are presented in anticipation of significant additions to the research.

B. CONCLUSIONS

This research demonstrates the feasibility of producing a CQB virtual training device that is both simple to use and deployable aboard a Navy vessel. The VCGDT, while still in its primitive stages, is an example of a system designed specifically around the constraints faced by its intended user. This 'made by Marines for Marines' concept is a solid first step toward filling a critical training gap in the Marine Corps' CQB mission. This is possible by moving ahead with the idea that virtual reality technology can assist in the training process only if implemented appropriately. Ideally, the research presented here has suggested that a fine line exists between a virtual environment that benefits training and one that

degrades it. It's this line that must be walked carefully in future work in this area.

C. FUTURE WORK

1. Vestibular Cues

Sensory cues commonly associated with human movement would be an interesting and beneficial addition to this research. Although studies in this field have yet to yield the ability to build the absolute model, many have shown positive results. Combining concepts from these studies with the VCGDT would need to account for cues associated with a ship borne environment. Although this could prove challenging, it's none-the-less an important concept to pursue.

2. Avatar Improvements

Limitations of the Boston Dynamics DI-Guy avatar were really limitations of time and resources. The avatars utilized in the prototype are all customizable to a fairly high degree. An example of this is the forward pointing feature of the instructor avatar. The difficulty in manipulating these characters to custom poses and animations is that the developer does it through a time-consuming and resource inefficient manner. Each of the body parts must be manipulated through a series of custom function calls, wasting both network resources and memory. Boston Dynamics has offered to customize the animations of their avatars, however, the costs are much greater than is feasible for research at this institution.

The answer is an avatar library built through our own department. The avatars used by our own set of classes can then be manipulated through custom animations by our own

modelers. This allows us to achieve any of the postures and gestures required by a CQB environment, without the high overhead associated with a licensed software application. This work is currently underway.

3. Audio improvements

Although the representation of sound in the VCGDT was a primary consideration with regard to immersion, not enough time was available to pursue optimizing this aspect. The gfAudio library's functionality extends far beyond its basic implementation in this prototype and incorporation of these advanced features would improve it immensely. Research is needed to determine which sounds and audio characteristics are important to the context of CQB missions and what types of hardware and software are needed to implement them.

4. Dynamic Scenarios

All of the targets, hostile or otherwise, are designed to be static in nature. No scripted avatar animation or motion was planned, nor were any intelligent agents introduced. An important research topic for future introduction would be uncovering what types of target movement and actions are beneficial to training. In conjunction with custom avatar work, these characteristics could be introduced to challenge user decision-making based on predetermined training objectives.

5. Prototype Testing

An obvious future item is evaluation. Since the VCGDT prototype's completion date left little time for robust experimentation, it's important that its effectiveness be tested in follow-on phases. An ideal method of

experimentation should include the use of a prototype in a ship borne environment. Evaluation can range from determining the effectiveness of the hardware aboard ship, the graphical presentation, audio aspects, and human factors like cyber-sickness.

LIST OF REFERENCES

[1] Smith, Roger D., "Essential Techniques for Military Modeling & Simulation," *Proceedings of the 1998 Winter Simulation Conference*. Orlando, Florida.

[2] Bell, Herbert H., "The Effectiveness of Distributed Mission Training," *Communications of the ACM*, September 1999, Vol.42, No.9.

[3] Macedonia, Michael, "Games Soldiers Play," *IEEE Spectrum*, March 2002.

[4] Coalescent Technologies distributed Virtual Training Environment Homepage, <http://www.ctcorp.com/performance15.html> (September 2002)

[5] Bachmann, Eric R., "Inertial and Magnetic Angle Tracking of Limb Segments for Inserting Humans into Synthetic Environments," Department of Computer Science, Naval Postgraduate School, Monterey, California, December 2000.

[6] Aronson, Warren, "A Cognitive Task Analysis of Close Quarter Battle," Naval Postgraduate School, Monterey, California, September 2002.

[7] Grossman, Dave, Lt.Col., "On Killing," Little, Brown and Company, 1996.

[8] Krebs, Eric, and Reece, Jordan, "GFLIB: 3D Graphics API Usability," Naval Postgraduate School, Monterey, California, March 2002.

[9] Darken, R.P., Cockayne, W.R., & Carmein, D., "The Omni-Directional Treadmill: A Locomotion Device for Virtual Worlds," *Proceeding of ACM UIST*, 1997, pp. 213-221.

[10] Couvillion, Warren, Lopez, Roger & Ling, Jian, " The Pressure Mat: A New Device for Traversing Virtual Environments Using Natural Motion," *Inter-service/Industry Training, Simulation and Education Conference*, May 2001, Orlando, Florida.

[11] Peterson, B., Wells, M., Furness, T.A. III, & Hunt, E., "The Effects of the Interface on Navigation Virtual Environments," *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting*, 1997, pp. 1496-1500, Santa Monica, California: Human Factors and Ergonomics Society.

[12] Paradiso, J., Hu, E., & Hsiao, K.-Y., "The CyberShoe: A Wireless Multisensor Interface for a Dancer's Feet," *Proceedings of International Dance and Technology 99*, Tempe, AZ; FullHouse Publishing, Columbus, Ohio (2000), pp. 5760.

[13] Harris, L., Jenkin, M. & Zikovitz, D.C., "Vestibular Cues and Virtual Environments," *Proceedings of the Virtual Reality Annual International Symposium*, 1998, pp. 133-138.

[14] Bowman, Doug A., Koller, David, & Hodges, Larry F., "Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques," *Proceedings of the Virtual Reality Annual International Symposium*, 1997, pp. 45-52.

[15] Singer, M.J., Ehrlich, J., Cinq-Mars, S., & Papin, J., "Task Performance in Virtual Environments: Stereoscopic Versus Monoscopic Displays and Head-Coupling," *U.S. Army Research Institute Technical Report*, 1995.

[16] DiZio, P. & Lackner, J.R., "Spatial Orientation, Adaptation, and Motion Sickness in Real and Virtual Environments," *Presence: Teleoperators and Virtual Environments 1(3)*, 1992, pp. 319-328.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Chairman, Code CS
   Computer Science Department, Code CS
   Naval Postgraduate School
   Monterey, California

4. Dr. Rudy Darken
   Modeling of Virtual Environment and Simulations
   (MOVES) Program
   Naval Postgraduate School
   Monterey, California

5. CDR Joseph Sullivan
   Computer Science Department, Code CS
   Naval Postgraduate School
   Monterey, California

6. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

7. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

8. Director, Marine Corps Research Center, MCCDC
   Code C40RC
   Quantico, Virginia

9. Marine Corps Tactical Systems Support Activity
   (Attn: Operations Officer)
   Camp Pendleton, California